

# Completion API for OCN

---

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	The Benefits of Completion API for OCN.....	3
1.2	Prerequisites to Completion API for OCN .....	3
<b>2</b>	<b>Calling Completion API for OCN .....</b>	<b>4</b>
2.1	Authentication.....	4
2.1.1	Obtaining an OAuth Token.....	4
2.2	Invoking Completion Rest API .....	9

---

# 1 Introduction

As a customer of SuccessFactors Learning, you can purchase a content subscription from an Open Content Network (OCN) provider. Your administrators can make the courses available to your users, and your users can launch the courses from their catalogs and learning assignments. Although the link to the content resides inside SuccessFactors Learning, the content can be hosted at the provider's site or a provider can have the content creators host their own content.

This API gives OCN partners a way to provide Learning completion information back to the customer's LMS. As a customer of SuccessFactors Learning, when you complete an Open Content Network (OCN) Item at the OCN-provider site, a completion can be automatically recorded for you in the Learning Management System (LMS).

## 1.1 The Benefits of Completion API for OCN

The Item completion is automatically recorded for LMS users from OCN providers that may not publish their courses in one of the Learning communication standards (AICC, SCORM, etc.) accepted by the LMS.

## 1.2 Prerequisites to Completion API for OCN

Before you can utilize the completion API for OCN with SuccessFactors Learning, you should have:

- A license to SuccessFactors Learning
- An agreement with at least one content provider on the OCN – contact your SuccessFactors representative for a current list of providers
- Users must complete Learning at the OCN-provider site that they connected to from the LMS.

The OCN provider will need to configure the `Configure Client_ID`, `Client_Secret` and `Tenant URL` to call the API.

---

## 2 Calling Completion API for OCN

The OCN provider has to call record completion rest API by passing appropriate parameters in order to record the completion of a learning item. The API is protected with OAuth.

### 2.1 Authentication

Completion API is protected with OAuth, a widely accepted framework for limited machine-to-machine data sharing. Full log in credentials are not stored or shared on the client machine. Instead, the client and server share a token, allowing the client access to only a limited set of data and methods on the data. The process of gaining a token and its policy creates trust between the two servers. You can learn more about OAuth on its web site and our SuccessFactors Learning API Reference Guide.

An authentication token establishes trust and limited access between two machines. The two machines in this case are the SuccessFactors Learning server and your web service client application. Obtaining a token is a process that ensures that the machines are who they claim to be and that all parties agree to the access granted by the token. Assuming that the business agreements are in place, obtaining a token is a two-step process:

1. An administrator generates a client secret that the client uses to get OAuth tokens from SuccessFactors Learning.
2. Using the client secret, the web service client requests OAuth tokens.

OAuth tokens expire, so the web service client periodically uses the client secret to obtain new tokens.

#### 2.1.1 Obtaining an OAuth Token

Navigate to System Admin > Configuration > OAuth Token Server to generate an OAuth Token. You obtain an OAuth token at the beginning of every web service session by using the client secret that a SuccessFactors Learning administrator generated for you.

### Context

You obtain an OAuth token to verify that your client is authorized to communicate with SuccessFactors Learning.

## Procedure

1. Concatenate the client ID and secret, separated by a colon. For example, if the client is t1 and the secret is 52d85f24492ff918dcd1935a45a9d34acd79e249fe331753c6e0105b109e034a then the result is:

```
t1: 52d85f24492ff918dcd1935a45a9d34acd79e249fe331753c6e0105b109e034a
```

2. Encode the concatenated client ID and secret with base64 encoding.  
For example, the following concatenation:

```
t1: 52d85f24492ff918dcd1935a45a9d34acd79e249fe331753c6e0105b109e034a
```

Encodes to:

```
dDE6NTJkODVmJjQ0OTJmZjkkOGRjZDE5MzVhNDVhOWQzNGFjZDc5ZTI0OWZlMzMxNzUzYzZlMDEwNWlzM  
D11MDM0YQ==
```

3. Over HTTPS, call the token granting service.  
Sample call to the token granting service:

```
POST {root}/learning/oauth-api/rest/v1/token
```

4. Pass the encoded client ID and client secret in the request header.

Sample header:

```
Authorization: Basic <encoded token from previous step>
```

SuccessFactors Learning web services follow HTTP 1.1 standard, so for all other HTTP header parameters, use them to industry best practices and HTTP 1.1 standards. For example, set content-type to application/json.

5. Pass required OAuth token parameters.
6. Check the response header, which is either 200: OK or an error with the HTTP error code.
7. Store the information in response body for subsequent calls. You must provide the authorization token in the header of subsequent calls, and the expiration and type information helps you to manage the lifecycle of the token.

## 2.1.1.1 Required OAuth Token Parameters

When you make a call to obtain an OAuth token from SuccessFactors Learning, these parameters are required. When you make a call to SuccessFactors Learning to obtain a token, you must pass the following parameters in the body of the request:

```
{
  "grant_type": "client_credentials",
  "scope": {
    "userID": "Usr1"
    "companyId": "t1"
    "userType": "user",
    "resourceType": "learning_public_api"
  }
}
```

Where:

grant\_type

For the scope of client web services, it is always client\_credentials.

scope

An object that describes the scope of access by a user that must exist on the specified tenant.

userID

The requested token's user ID.

companyId

The tenant ID, which you can find the tenant ID in a number of locations, including the page that generates the client ID and secret.

userType

If the user is an "of type" administrator, then the value is "admin" or "A." If the user is an of type user, then the value is "user" or "P." We recommend spelling the type out (admin or user) to make the type clear.

resourceType

For the scope of this document, it is always "learning-public-api."

---

The call returns:

```
{
  issuedAt:<epoch UTC>,
  expiresIn: <epoch UTC>,
  ,
  issuedFor: "learning_public_api",
  access_token: "<token>"
}
```

Where:

IssuedAt

The time, in Unix epoch, at which the token was issued.

expires\_in

The time, in Unix epoch, at which the token expires. Use this to understand when you need to call for another token.

IssuedFor

The issued resource's resource type. For the purpose of web services to SuccessFactors Learning, it is "learning\_public\_api."



---

## 2.2 Invoking Completion Rest API

In order to record completion, we have to call the following end point:

`https://(learning server)/learning/public-api/rest/v1/current-user/item/learning-event`

Here is an example of a sample request:



### RESOURCE DESCRIPTION

Request URL `https://(learning server)/learning/public-api/rest/v1/current-user/item/learning-event`

Request Method POST

Description API to record learning for a User

Since b1502

Request Header Accept: application/json

Parameters None

Request Body

```
{"userID": "<USERID>", "courseID": "<COURSEID>", "providerID": "<PROVIDERID>",  
"courseCompleted": "true", "completedTimestamp": "<COMPLETEDTIMESTAMP>", "instructorName": "  
<InstructorName>", "grade": "<grade>", "totalHours": "<TOTALHOURS>", "creditHours": "<CREDITHOUR  
>", "contactHours": "<CONTACTHOURS>", "cpeHours": "<CPE>", "comments": "<COMMENTS>", "price" :  
<PRICE>, "currency": "<CURRENCY>"}
```

RESPONSE HEADER 200: OK or ERROR see HTTP error codes.

RESPONSE Body Well formed JSON string which binds to RestOperationStatusVOX

---

# Document History

Version	Date	Change
1.0	2015-03-18	Document Created

# Typographic Conventions

Type Style	Description
<i>Example</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
<b>Example</b>	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<b>Example</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE	Keys on the keyboard, for example, F2 or ENTER.

